Beispiel 1

```
/* Dateiname: A.java
 * Programmiersprachen I, Hochschule Karlsruhe
 * Beispiel 1 zum Tutorium, Handzettel 1
 * Autor: Arne Johannessen
 * erstellt anhand einer Vorlage von Prof. Dr. B. Buerg
 * geschrieben am 2006-04-21
 */

public class A { public static void main (String args
[]) { int g = 10; for (int d = g; d > 0; d--) { for
(int u = 1; u < d; u++) { System.out.print(' '); } int
k = 2 * (g - d) + 1; for (int s = 0; s < k; s++) {
System.out.print('o'); } System.out.println(); } } }
```

Guter Code ist übersichtlich und selbsterklärend (Beispiel 2).

Einige der besprochenen Punkte:

- Einrückung ist in Java nicht nötig (wegen geschweifter Klammern), aber sinnvoll zur besseren Übersicht
- richtige Einrückung notwendig bei „natürlicher Sprache“
- immer nur ein Befehl pro Zeile
- Zeilen schließen in der Regel entweder mit Semikolon ; oder mit runder Klammer zu) ab
- Variablennamen sollten „sprechend“ sein, also dem tatsächlichen Inhalt der Variablen entsprechen
- Kommentare (`//…` oder `/*…*/`) sind erwünscht; sie sollen nicht dokumentieren, *was* geschieht, sondern *warum* es geschieht
- Strategie zur Aufstellung von Algorithmen: „Top-Down-Verfahren“ – dabei werden erst einige wenige Schritte stark abstrahiert formuliert (z. B. „Einrückung für Baum erzeugen“), bevor der eigentliche Code für diese Schritte geschrieben wird.
- auch ausgehend vom Algorithmus in „natürlicher Sprache“ kann Java-Code vergleichsweise einfach erstellt werden

Beispiel 2

```
/* Dateiname: Baum.java
 * Programmiersprachen I, Hochschule Karlsruhe
 * Beispiel 2 zum Tutorium, Handzettel 1
 * Autor: Arne Johannessen
 * erstellt anhand einer Vorlage von Prof. Dr. B. Buerg
 * geschrieben am 2006-04-21
 */

/**
 * Zeichnet einen aufrecht stehenden dreieckigen Baum, der
 * aus 'o's besteht. Der Baum hat eine Hoehe von 10 Zeilen.
 */
public class Baum
{

    public static void main (String args [])
    {

        // Hoehe des Baums: 10 Zeilen (vorgegeben)
        int hoehe = 10;

        // den Baum Zeile fuer Zeile zeichnen
        for (int zeile = hoehe; zeile > 0; zeile--)
        {

            // Einrueckung fuer Baum erzeugen
            for (int randSpalte = 1; randSpalte < zeile; randSpalte++)
            {
                System.out.print(' ');
            }

            // eigentlichen Baum erzeugen
            int zeilenBreite = 2 * (hoehe - zeile) + 1;
            for (int baumSpalte = 0; baumSpalte < zeilenBreite; baumSpalte++)
            {
                System.out.print('o');
            }

            System.out.println();
        }

    }

}
```